Visualizing Time-Varying Phenomena In Numerical Simulations Of Unsteady Flows

David A. Lane
NAS-96-001 February 1996

MRJ, Inc.
NASA Ames Research Center
M/S T27A-2
Moffett Field, CA 94035

`lane@nas.nasa.gov`
`http://www.nas.nasa.gov/NAS/TechReports/nas_96_001`

Also appeared as paper AIAA-96-0048 in the 34th Aerospace Sciences Meeting & Exhibit.

Abstract

Streamlines, contour lines, vector plots, and volume slices (cutting planes) are commonly used for flow visualization. These techniques are sometimes referred to as ***instantaneous*** *flow visualization techniques because calculations are based on an instant of the flow field in time. Although instantaneous flow visualization techniques are effective for depicting phenomena in steady flows, they sometimes do not adequately depict time-varying phenomena in unsteady flows. Streaklines and timelines are effective visualization techniques for depicting vortex shedding, vortex breakdown, and shock waves in unsteady flows. These techniques are examples of* ***time-dependent*** *flow visualization techniques, which are based on many instants of the flow fields in time.*

This paper describes the algorithms for computing streaklines and timelines. Using numerically simulated unsteady flows, streaklines and timelines are compared with streamlines, contour lines, and vector plots. It is shown that streaklines and timelines reveal vortex shedding and vortex breakdown more clearly than instantaneous flow visualization techniques.

Introduction

Numerical simulations of unsteady flows are becoming feasible because of advances in computing and hardware capabilities, yet there are still relatively few visualization techniques developed specifically for numerically-simulated unsteady flows. For steady flow visualization, there are several well-known techniques. Streamlines are curves tangent to the instantaneous flow field. Contour lines are isocurves of some scalar field. Vector plots show the direction and magnitude of the velocity at the specified grid points. Volume slices, which are also referred to as cutting planes, display flow features at several grid planes or Cartesian planes. All of these techniques are computed based on the flow field at some instant in time.

For unsteady flow visualization, many time steps of the flow data are required. A common approach is to use instantaneous flow visualization techniques to visualize the flow at individual time steps. Thus, for a given time step, streamlines, contour lines, vector plots and/or volume slices are computed and sometimes saved for playback. By animating the streamlines, contour lines, and vector plots computed from each time step, one hopes to see the time-varying phenomena in the unsteady flow.

Although the approach described above is simple and commonly used, it is not always the most effective way to visualize unsteady flows. Phenomena such as vortex shedding, vortex breakdown, and shock waves evolve in time, and in order to depict these phenomena, time should be considered in the calculation. Instantaneous flow visualization techniques do not consider the fact that the flow is unsteady and calculation is performed independent of time. Streaklines and timelines are examples of unsteady flow visualization techniques that consider time in the calculation. A streakline is a curve formed by all particles that were previously injected from a fixed seed location. A timeline is a curve formed by all particles that were injected simultaneously at some instant in time from a set of fixed seed locations. Both of these techniques trace particles through many time steps of the given unsteady flow data.

In this paper algorithms for computing streaklines and timelines are described. Comparisons of instantaneous and time-dependent flow visualization techniques are made.

Related Work

There is published literature on several instantaneous flow visualization techniques. Some of these techniques are surveyed in [1,2,3]. Presently, there are only a few unsteady flow visualization systems and most of them were developed for 2D unsteady flows. In [4], 2D unsteady flows were computed for an oscillating airfoil. Streamlines and streaklines were computed to visualize the flow. It was warned that observing instantaneous streamlines for unsteady flows could be misleading and that streaklines should be used instead. Simulated streaklines have been shown to be useful in analyzing photographs of unsteady flows [5]. Several photographs of streaklines from actual experiments were compared with simulated streaklines. 3D unsteady flows past a tapered cylinder were simulated and visualized [6]. Streaksurfaces were computed by releasing thousands of particles along a 2D rake at each time step. In [7], streaklines were computed to validate experimental flow visualization techniques such as smoke. It was observed that the placement of the seed points of the streaklines is critical in obtaining a good depiction of the flow. We have developed a particle tracing system for 3D unsteady flows [8]. It supports multi-zoned curvilinear moving grids. A vortex core tracing algorithm for visualizing unsteady 2D vortices was introduced in [9]. The algorithm was enhanced for 3D unsteady problems by including time as the third dimension. Once the vortex core has been computed, it is then represented by a vortex tube using a local tilting method.

Streaklines

In experimental flow visualization, the behavior of the flow is analyzed by continuously injecting smoke or dye into the flow from some fixed seed locations. The smoke or dye is then photographed after some time has elapsed. In hydrodynamic experiments, dyes instead of smoke are used. If the flow is steady, then the smoke patterns photographed are streamlines. Several illustrative photographs of smoke/dye taken from experimental flows can be found in [10].

In numerical flow simulations, the flow data are saved only at discrete points in time and space. To simulate streaklines, small massless particles are injected into the flow from several seed locations at each saved time step. The particles are then traced both spatially and temporally. The grid used to set up the flow equations may be of several types; for example, rectilinear, cartesian, or curvilinear grids. If the grid is rectilinear or cartesian, then particle tracing is simple and fast. However, if the grid is curvilinear then particle tracing becomes nontrivial because of point location (cell search) and velocity transformation (from physical space to computational space). There are also additional complexities when the grid consists of multiple blocks (a multi-zoned grid). Furthermore, if the grid moves in time (unsteady grid), then particle tracing requires special handling. For a detailed discussion of particle tracing in multi-zoned curvilinear moving grids, see [11]. The algorithms described below are independent of the grid type and whether particle tracing is performed in physical or computational space.

Assume that the velocity field from the numerically simulated flow is available for time *t_0,...,t_n*, where *n* is the number of saved time steps, and at distinct grid points. Let *p(t)* be the position of the particle at time *t* and *v(p(t),t)* be the velocity of the particle at *p(t)* at time t, then after time Δt, the new position of the particle becomes:

$$p(t+\Delta t) = p(t) + \int_t^{t+\Delta t} v(p(t), t)\, dt \quad (1)$$

Equation (1) can be evaluated by numerical integration. Using the fourth order Runge-Kutta scheme, (1) becomes:

$$p(t+\Delta t) = p(t) + (a + 2b + 2c + d)/6, \quad (2)$$

where

$$a = \Delta t v(p(t), t), \; b = \Delta t v(p(t) + a/2, t + \Delta t/2),$$

$$c = \Delta t v(p(t) + b/2, t + \Delta t/2),$$

$$d = \Delta t v(p(t) + c, t + \Delta t),$$

$$t = t + \Delta t.$$

Thus, using equation (2), a particle can be numerically integrated and traced through the flow field.

Let $s_j, j=0,...,m$ be the user-specified seed locations. To generate simulated streaklines, particles are injected from these seed locations at each time step. Once injected, the particles are traced in time until they exit the grid. Two methods can be used to perform this task. The first method is to trace particles individually from time t_0 to t_n. Thus, particles from seed s_j are traced through all time steps before tracing particles from seed s_{j+1}. An alternative and more efficient method is to trace particles from all seeds at each time step. This is repeated until all of the time steps have been used or until all particles have exited the grid. In this method, all particles are traced from time t_i to t_{i+1} before advancing to time step t_{i+2}. The first method is simple, but it requires accessing the same time step of the flow field repeatedly for each particle. If the unsteady flow dataset is small such that all time steps of the flow can be stored in memory, then this method would work adequately. However, if the flow dataset is too large to be loaded into memory, which is often the case for 3D unsteady flows, then there is an overhead for repeatedly reading the same time step of the flow data. The second method only requires accessing each time step's data once since all particles are traced ``simultaneously'' in the same time step interval. Though there is a small memory overhead for saving the active particles at each time step, the memory overhead is usually smaller than one time step of the flow data. Following is a procedure for computing streaklines using the second method:

1. Read the seed locations $s_j, j=0,...,m$.

2. Inject one particle from each seed s_j.

3. Read the flow data for time t_0.

4. For $i=1,...,n$ do the following steps:

    4.1. Read the flow data for time t_i.

    4.2. Trace all active particles from time t_{i-1} to t_i using equation (2).

    4.3. Inject one particle from each seed s_j.

    4.4. Assign color to all active particles based on the specified scalar field.

    4.5. Render all active particles.

For playback purpose, Step 4.5 above can be modified such that all active particles at each time step are saved to a graphics metafile, which then can be rendered repeatedly at a later session.

When comparing experimental streaklines with numerical streaklines, experimental streaklines are usually smooth and continuous, whereas numerical streaklines are represented by discrete points. This is because particles are injected continuously in experimental flow; however, in numerical flows, particles are injected at discrete time steps. The time resolution of the numerical flow dictates how continuous the streaklines appear. Unfortunately, most large 3D unsteady flow simulations are saved at a coarse resolution, for example, at every 50th or 100th time step due to disk limitation. Hence, numerical streaklines may not always be smooth. For this reason, numerical streaklines usually are not connected because they can be jagged. Nevertheless, numerical streaklines still can reveal features that are visible in experimental streaklines.

Timelines

In experimental flow visualization, timelines are sometimes generated. By calculating the distance between neighboring timelines, one can determine the velocity of the particles in the flow. Rows of particles are injected intermittently into the flow to form timelines. Each row has a very high density of particles. Thus, the number of seed locations (m) should be large. In experimental flow, m is infinite because of the precision of the hardware; hence, timelines have continuous form. Numerical timelines usually do not have continuous form because the smoke is represented by discrete particles. If the neighboring particles are closely clustered, then simulated timelines will look continuous. However, they can become fragmented when neighboring particles are separated.

To compute timelines, the procedure shown in the previous section needs slight modification. Instead of releasing particles at each time step as indicated by Step 4.3, particles are injected at a specified time interval. For example, at every 5th or 10th time step. Furthermore, the number of seed points along a rake is usually larger than the number of seed points used for computing streaklines.

Comparisons

In this section three commonly used instantaneous flow visualization techniques: vector plots, contour lines, and streamlines are compared with streaklines and timelines. In the following figures, PLOT3D [12] was used to generate the vector plots and contour lines, and the Unsteady Flow Analysis Toolkit (UFAT) was used to generate instantaneous streamlines, streaklines, and timelines. UFAT which was developed specifically for unsteady flow visualization has been used to visualize several large 3D unsteady flows with multi-zoned curvilinear moving grids. For a detailed discussion of UFAT and its features, see [13].

A 2D oscillating airfoil is used for the first comparison. Unsteady flow about the 2D airfoil was simulated for three flow conditions: attached flows, light dynamics stall, and deep dynamic stall [14]. The simulated flow was compared to the experimental data.

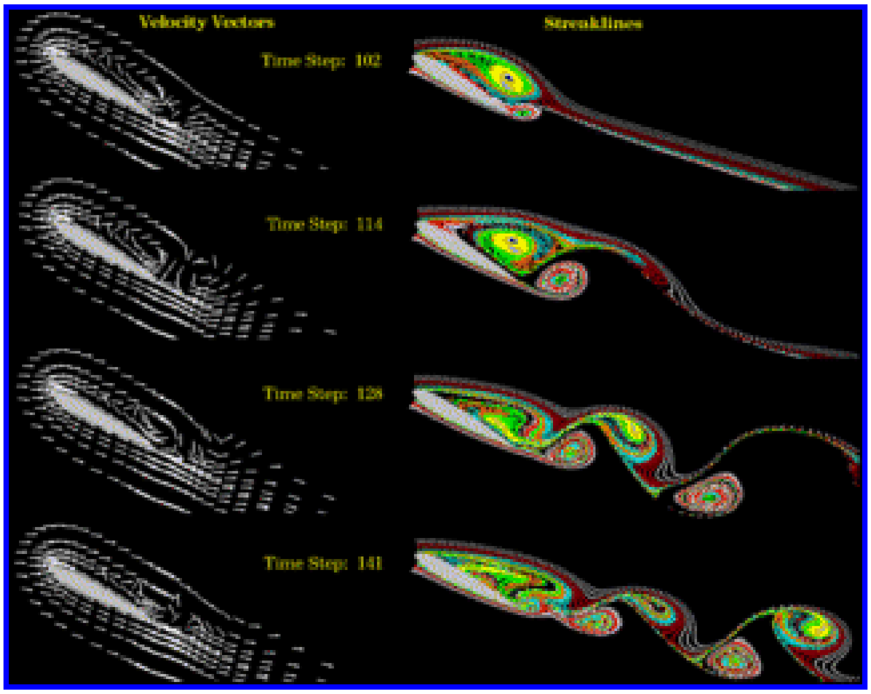

Figure 1 (Click on image for larger view)

Figure 1 depicts velocity vector plots and streaklines at four time steps taken from an animation of the simulated flow. It can be seen from the figure that the vector plots do not reveal vortices that are clearly depicted in the streaklines.

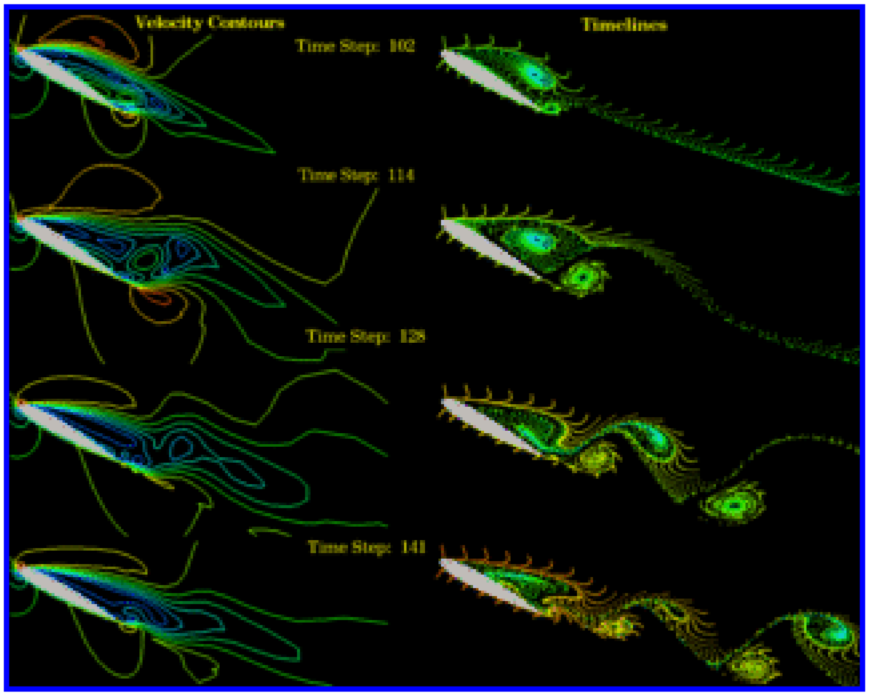

Figure 2 (Click on image for larger view)

Figure 2 shows velocity magnitude contours and timelines. Although the contours shown give some indication of the existence of vortices in the flow, the timelines shown give a better depiction of the vortices.

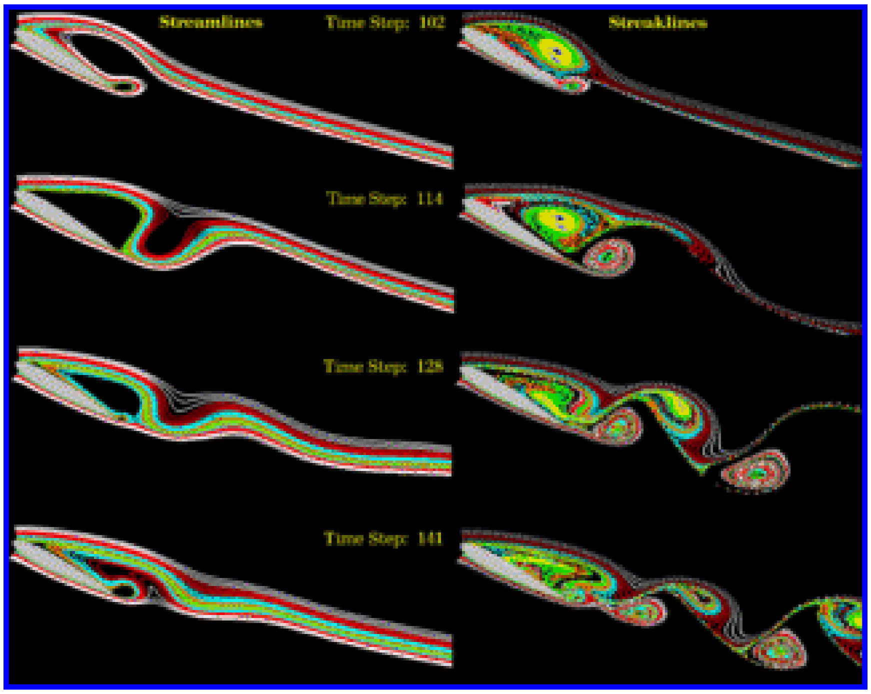

Figure 3 (Click on image for larger view)

In Figure 3, instantaneous streamlines and streaklines are compared. Streamlines are commonly used to depict the flow at an instant in time. However, as shown in the figure, streamlines sometimes do not reveal vortex development and shedding clearly. Streaklines shown in the figure give a better depiction of these time- varying physical phenomena. The differences between instantaneous and time- dependent techniques are even more apparent when the flow is animated.

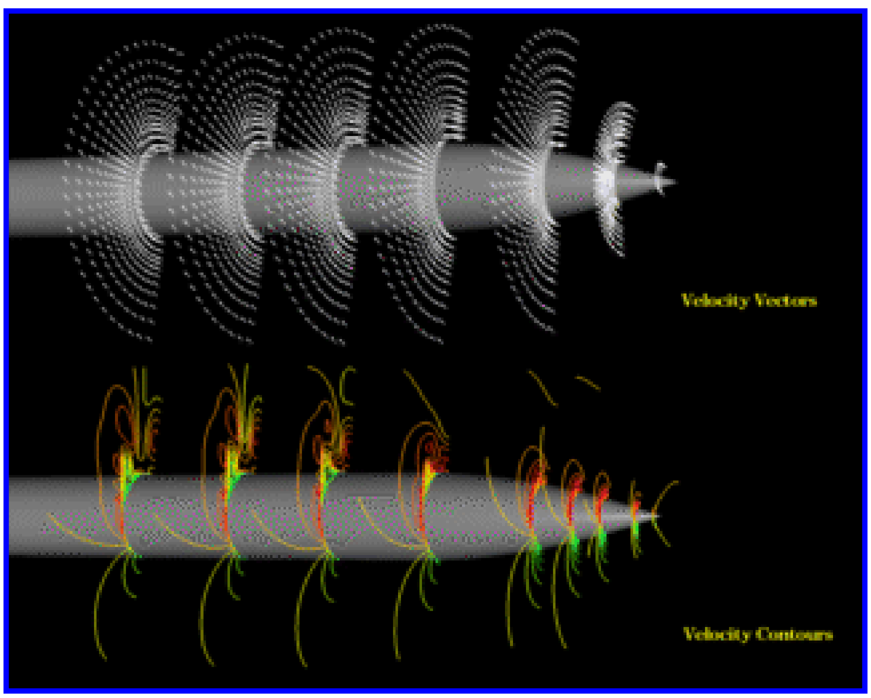

Figure 4 (Click on image for larger view)

Flow simulation of an ogive cylinder was used to compare instantaneous and time-dependent flow visualization techniques. Figure 4 shows velocity vector plots and contours of velocity magnitude at several cross sections along the ogive cylinder. The vector plots do not reveal any vortex structure in the flow, which is misleading because there are some vortex structures in the flow. The velocity contours give some indications of the vortex structures, but not completely. Both techniques are instantaneous techniques.

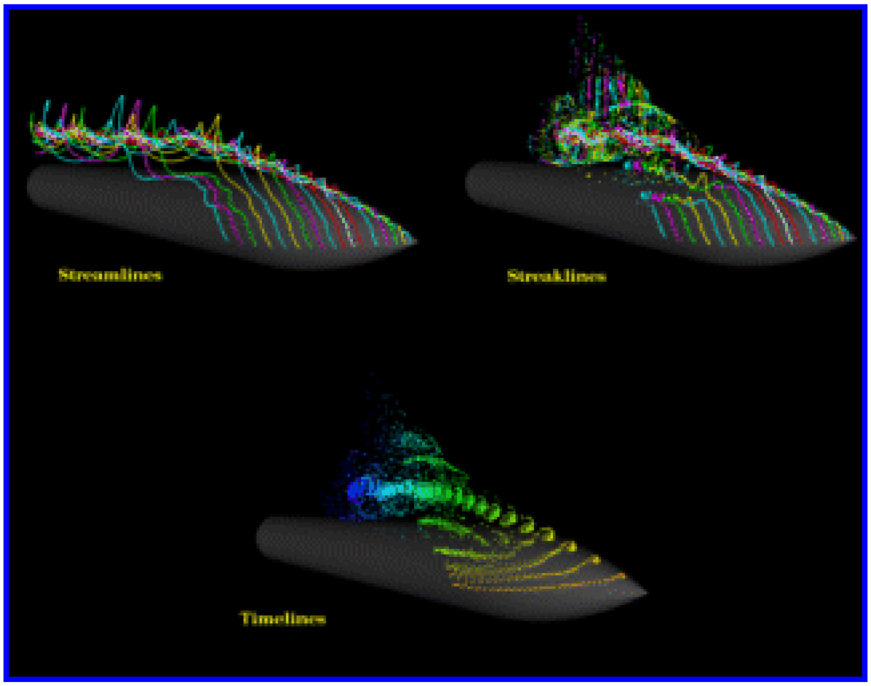

Figure 5 (Click on image for larger view)

Figure 5 depicts streamlines, streaklines and timelines. All three techniques give the indication of the primary vortex moving along the body of the cylinder. However, both streaklines and timelines give a much better indication of vortex shedding and formation when animated.

Conclusions

Instantaneous flow visualization techniques such as streamlines, velocity plots, and contour lines are commonly used for unsteady flow visualization. However, they may not be accurate for depicting unsteady flows because the time variable is not considered in the calculation. Time- dependent flow visualization techniques such as streaklines and timelines are effective particle tracing methods for depicting time-varying phenomena in unsteady flows. For an accurate analysis, streaklines and timelines should also be used to complement the instantaneous techniques.

Acknowledgments

The airfoil data set was provided by Sungho Ko and Scott Murman provided the ogive cylinder data set. This work was supported by NASA under contract NAS 2-14303.

References

[1] Buning, P., Numerical algorithms in CFD post-processing, Computer Graphics and Flow Visualization in Computational Fluid Dynamics, von Karman Institute for Fluid Dynamics Lecture Series 1989-07, September, 1989.

[2] Strid, T., Rizzi, A., and Oppelstrup, J., Development and use of some flow visualization algorithms, Computer Graphics and Flow Visualization in Computational Fluid Dynamics, von Karman Institute for Fluid Dynamics Lecture Series 1989-07, September, 1989.

[3] Post, F. and van Wijk, J., Visual representation of vector fields: recent developments and research directions, Scientific Visualization Advances and Challenges, Academic Press, San Diego, 1993, pp. 367-390.

[4] Barth, T., Pulliam, T., and Buning, P., Navier-Stokes computations for exotic airfoils, AIAA 23rd Aerospace Sciences Meeting, Reno, Nevada, January 1985, AIAA 85-0109.

[5] Chow, C., Leben, R., and Gea, L., Numerical simulation of streaklines in unsteady flows, AIAA-89-0292.

[6] Jaspersen, D. and Levit, C., Numerical simulation of flow past a tapered cylinder, AIAA 29th Aerospace Sciences Meeting, Reno, Nevada, January 1991, AIAA 91-0751.

[7] Shariff, K., Pulliam, T., and Ottino, J., A dynamical systems analysis of kinematics in the time-periodic wake of a circular cylinder, Lectures in Applied Mathematics, Vol. 28, 1991.

[8] Lane, D., Visualization of time-dependent flow fields, Proceedings of Visualization `93, San Jose, California, October 1993, pp. 32-38.

[9] Ma, K. and Zheng, Z., 3D visualization of unsteady 2D airplane wake vortices, Proceedings of Visualization `94, Washington, D.C., October 1994, pp. 257-264.

[10] Van Dyke, M., An Album of Fluid Motion, Parabolic Press, Stanford, California, 1982.

[11] Lane, D., Scientific visualization of large-scale unsteady fluid flow, to appear in: G. Nielson, H. Hagen, H. Mueller (eds.), Scientific Visualization: Surveys, Methodologies and Techniques, IEEE Computer Society Press, 1996.

[12] Buning, P. and Steger, J., Graphics and flow visualization in computational fluid dynamics, 7th Computational Fluid Dynamics Conference, Cincinnati, Ohio, July 1985, AIAA 85-1507.

[13] Lane, D., UFAT -- a particle tracer for time-dependent flow fields, Proceedings of Visualization `94, Washington, D.C., October 1994, pp. 257-264.

[14] Ko, S. and McCrosky, W., Computations of unsteady separating flows over an oscillating airfoil, AIAA 33rd Aerospace Sciences Meeting, Reno, Nevada, January 1995, AIAA 95-0312.